2009-10-01

An Improved CamShift Algorithm for Target Tracking in Video Surveillance

Chunrong Zhang
Athlone Institute of Technology

Yuansong Qiao
Athlone Institute of Technology

Enda Fallon
Athlone Institute of Technology

Chiangqiao Xu
Chinese Acadmey of Sciences, Beijing

An Improved CamShift Algorithm for Target Tracking in Video Surveillance

Chunrong Zhang[1, 2, 3], Yuansong Qiao[1,2], Enda Fallon[1], Changqiao Xu[2]

[1] Software Research Institute, Athlone Institute of Technology, Athlone, Ireland
[2] Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]Graduate University of the Chinese Academy of Sciences, Beijing, China
crzhang08@gmail.com, {ysqiao, efallon}@ait.ie, cqiaoxu@gmail.com

Abstract

Target tracking in a cluttered environment remains a challenging research topic. The task of target tracking is a key component of video surveillance and monitoring systems. In this paper, we present an improved CamShift algorithm for tracking a target in video sequences in real time. Firstly, a background-weighted histogram which helps to distinguish the target from the background and other targets is introduced. Secondly, the window size is calculated to track the target as its shape and orientation change. Finally, we use a Kalman Filter to avoid being trapped by a local maximum. The introduction of the Kalman Filter also enables track recovery following a total occlusion. Experiments on various video sequences illustrate the proposed algorithm performs better than the original CamShift approach.

Key words: Target tracking; CamShift ; Kalman filter ; Background-weighted histogram

1 Introduction

Network video surveillance has been a popular security application for many years. Target tracking in a cluttered environment remains one of the challenging problems of video surveillance. The task of target tracking is a key component of video surveillance and monitoring systems [1]. It provides input to high-level processing such as recognition [2], access control, or re-identification, or is used to initialize the analysis and classification of human activities.

Tracking algorithms can be classified into two major groups, namely state-space approach and kernel-based approach. State-space approaches are based largely on probability, stochastic processes and estimation theory, which, when combined with systems theory and combinatorial optimization, lead to

a plethora of approaches, such as Kalman filter, Extended Kalman Filter (EKF) [3], Unscented Kalman Filter (UKF)[4], Particle Filter (PF)[5]. The ability to recover from lost tracks makes State-space approach one of the most used tracking algorithms. However, some of them require high computational costs so they are not appropriate for real time video surveillance systems.

The Mean Shift (MS) algorithm is a non-parametric method which belongs to the second group. MS is an iterative kernel-based deterministic procedure which converges to a local maximum of the measurement function under certain assumptions about the kernel behaviors [6]. CamShift (Continuously Adaptive Mean Shift) algorithm [7] is based on an adaptation of mean shift that, given a probability density image, finds the mean (mode) of the distribution by iterating in the direction of maximum increase in probability density. CamShift algorithm has recently gained significant attention as an efficient and robust method for visual tracking. A number of attempts have been made to achieve robust, high-performance target tracking [8][9][10].

CamShift algorithm is a low complexity algorithm, which provides a general and reliable solution independent of the features representing the target. But it has some important inherent drawbacks. Firstly , the algorithm may fail to track multi-hued targets or targets where hue alone cannot allow the target to be distinguished from the background and other targets. Secondly, CamShift is primarily intended to perform efficient head and face tracking in a perceptual user interface, it may lose the target when the target's shape and orientation are changing. Thirdly, CamShift, like the mean shift algorithm, can only be used to find local modes [11]. It fails in tracking small and fast moving targets (interframe displacement larger than their size) because it is trapped in a local maximum. Finally, for single stationary camera surveillance, target occlusion is a common phenomenon owing to the limitation of camera views. CamShift cannot track the target when a total occlusion happens.

The algorithm proposed here is using a tracker representing the center of the target. The tracker tracks the target by the CamShift algorithm. Then, to avoid being trapped by a local maximum, we search the true maximum beyond the local one by using the Kalman Filter. In the meantime, the Kalman Filter can also help to recover a track after a total occlusion. In addition, we use the background-weighted histogram to distinguish the target from the background.

The rest of the paper is organized as follows: Section 2 presents the original mean shift and CamShift algorithms. The proposed tracking algorithm is developed and analyzed in Section 3. Experiments and comparisons are given in Section 4, and the conclusion is in Section 5.

2 The Original CamShift Algorithm

2.1 Mean Shift Algorithm

The mean-shift algorithm is a non-parametric density gradient estimator. It is basically an iterative expectation maximization clustering algorithm executed within local search regions. Comaniciu has adapted the mean-shift for the tracking of manually initialized targets [12]. The mean-shift tracker provides accurate localization and it is computationally feasible.

A widely used form of target representation is color histograms, because of its independence from scaling and rotation and its robustness to partial occlusions. Define the target model as its normalized color histogram, $q = \{q_u\}_{1,\ldots m}$,

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|x_i^*\|^2)\, \delta[b(x_i^*) - u] \tag{1}$$

where m is the number of bins. The normalized color distribution of a target candidate $p(y) = \{p_u(y)\}_{1,\ldots nh}$ centered in y can be calculated as

$$p_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \tag{2}$$

where $\{x_i\}, i=1,\ldots n_h$ are the n_h pixel locations of the target candidate in the target area, $b(x_i)$ associates the pixel x_i to the histogram bin, $k(x)$ is the kernel profile with bandwidth h, and C_h is a normalization function defined as

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \tag{3}$$

In order to calculate the likelihood of a candidate we need a similarity function which defines a distance between the model and the candidate. A metric can be based on the Bhattacharyya coefficient [13], defined between two normalized histograms $p(y)$ and q as

$$\rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y), q_u} \tag{4}$$

Hence we define the distance as

$$d[p(y), q] = \sqrt{1 - \rho[p(y), q]} \tag{5}$$

To track the target using the Mean Shift algorithm, it iterates the following steps:

1. Choose a search window size and the initial location of the search window.
2. Compute the mean location in the search window.
3. Center the search window at the mean location computed in Step 2.
4. Repeat Steps 2 and 3 until convergence (or until the mean location moves less than a preset threshold).

2.2 CamShift Algorithm

In the CamShift Algorithm, a probability distribution image of the desired color in the video sequence is created. It first creates a model of the desired hue using a color histogram and uses the Hue Saturation Value (HSV) color system [14] that corresponds to projecting standard RGB color space along its principal diagonal from white to black. Color distributions derived from video image sequences change over time, so the mean shift algorithm has to be modified to adapt dynamically to the probability distribution it is tracking.

CamShift is primarily intended to perform efficient head and face tracking in a perceptual user interface. For face tracking, CamShift tracks the X, Y, and Area of the flesh color probability distribution representing a face. Area is proportional to Z, the distance from the camera. Head roll is also tracked as a further degree of freedom. Then Bradski [7] uses the X, Y, Z, and Roll derived from CamShift target tracking as a perceptual user interface for controlling commercial computer games and for exploring 3D graphic virtual worlds.

CamShift algorithm is based on an adaptation of mean shift algorithm. And it is calculated as:

1. Choose the initial location of the search window.
2. Mean Shift as above (one or many iterations); store the zeroth moment.
3. Set the search window size equal to a function of the zeroth moment found in Step 2.
4. Repeat Steps 2 and 3 until convergence (mean location moves less than a preset threshold).

For discrete 2D image probability distributions, the mean location (the centroid) within the search window can be found by the zeroth moment. The window size, *s*, can also be set by the zeroth moment. The 2D orientation of the probability distribution is also easy to obtain by using the second moments, and then length, *l*, and width, *w*, of the target can be calculated.

3 The Proposed Algorithm

In this paper, we present an improved CamShift algorithm to solve the problems in original CamShift algorithm. Firstly, a background-weighted histogram which helps to distinguish the target from the background and other targets is introduced. Secondly, the window size is calculated to track the target as its shape and orientation change. Finally, we use a Kalman Filter to avoid being trapped by a local maximum. By combining the CamShift and Kalman Filter, we propose a real time tracking algorithm which copes with a temporal occlusion with a small computational cost. Fig. 1 summarizes the algorithm. The proposed algorithm is based on the original CamShift algorithm. To avoid the drawbacks we add several modules to improve the target tracking performance.

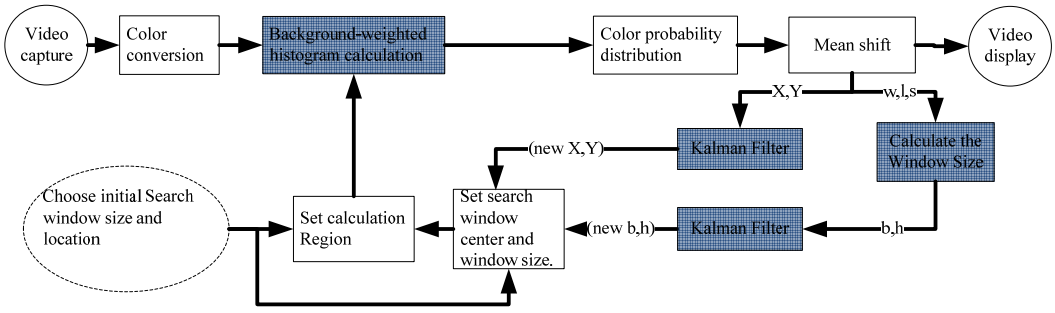

Fig. 1 Flowchart of the Proposed Algorithm

3.1 Background-weighted histogram

The background information is important for target tracking. Let $\hat{o}_u$ be the discrete representation (histogram) of the background in the feature space. We define the background histogram, which is a discrete un-weighted representation of a significant region outside the target region:

background: $\hat{o} = \{\hat{o}_u\}_{u=1\ldots m}$ $\sum_{u=1}^{m} \hat{o}_u = 1$ (6)

From the background model a set of weights are defined that allow the significance of certain unitized features to be diminished in the target and candidate model. We use the standard derivation of the weight where $\hat{o}^*$ is the smallest non-zero entry selected from the background model:

feature weights: $$\left\{v_u = \min\left(\frac{\hat{o}^*}{\hat{o}_u}, 1\right)\right\}_{u=1\ldots m} \quad (7)$$

These weights are employed to define a transformation for the representations of the target model and candidates. The transformation diminishes the importance of those features which have low v_u , i.e., are prominent in the background. Compare with (1), the new target model representation is then defined by

$$\hat{q}_u = C v_u \sum_{i=1}^{n} k(\|x_i^*\|^2)\, \delta[b(x_i^*) - u] \quad (8)$$

with the normalization constant C expressed as

$$C = \frac{1}{\sum_{i=1}^{n} k\left(\|x_i^*\|^2\right) \sum_{u=1}^{m} v_u \delta[b(x_i^*) - u]} \quad (9)$$

Compare with (2) and (3), similarly, the new target candidate representation is

$$p_u(y) = C_h v_u \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (10)$$

where now C_h is given by

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \sum_{u=1}^{m} v_u \delta[b(x_i) - u]} \quad (11)$$

3.2 Calculate the Search Window Size

In CamShift, the size *s* of the search window can be found. For tracking faces, Camshift sets window width to *s* and window length to 1.2*s* since faces are somewhat elliptical. But in other tracking systems, the accurate width and height of the window (ROI) are unknown. Also, the shape and orientation of the targets are changing.

To solve this problem, we calculate the width and the height of the search window. Suppose the width is *b*, the height is *h*, the size is *s*, then:

$$b*h = s^2 \quad (12)$$

The search window should be proportional to the axis, so compute the length axis *l* and width axis *w* from the distribution centroid, we can get:

$$b/\,h = w/l \quad (13)$$

Then:

$$b = \sqrt{w/l} * s \qquad h = \sqrt{l/w} * s \quad (14)$$

Considering the target orientation θ, the width *bn* and the height *hn* of ROI can be found using the following formulae.

$$bn = (b * cos\,\theta + h * sin\,\theta) \qquad hn = (b * \sin\theta + h * \cos\theta) \quad (15)$$

Experimental results show that when θ is small, computing *b* and *h* is sufficient. For other values of θ, *bn* and *bh* are better.

3.3 Kalman filter

The Kalman filter algorithm belongs to the state-space approach class of tracking algorithms. It solves the tracking problem based on the state-space equation and the measurement equation. To avoid being trapped by a local maximum, we first use one Kalman Filter to search the true maximum beyond the local one. The Kalman Filter is used to locate the start point that CamShift will search. We define the state-space equation:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{x_{k+1}} \\ v_{y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix} + W_k \tag{16}$$

and the measurement equation:

$$\begin{bmatrix} x_{ck} \\ y_{ck} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix} + V_k \tag{17}$$

Where k≥1, W_k is a white Gaussian noise with diagonal variance Q, V_k is a white Gaussian noise with diagonal variance R. x_k, y_k is the centroid of the search window, x_{ck}, y_{ck} is the current measurement of the centroid. v_{x_k}, v_{y_k} is the velocity (displacement) of the target. T is the interval between the frames.

In addition, we use another Kalman Filter to predict the search window's width, b, and height h (15). We define the state-space equation:

$$\begin{bmatrix} b_{k+1} \\ h_{k+1} \\ r_{b_{k+1}} \\ r_{h_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} b_k \\ h_k \\ r_{b_k} \\ r_{h_k} \end{bmatrix} + U_k \tag{18}$$

and the measurement equation:

$$\begin{bmatrix} b_{ck} \\ h_{ck} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} b_k \\ h_k \\ r_{b_k} \\ r_{h_k} \end{bmatrix} + Z_k \tag{19}$$

where b_k, h_k is the width and height of the search window, b_{ck}, h_{ck} is the current measurement of the width and height. r_{b_k}, r_{h_k} is the ratio of scale of the windows which is proportional to the scale of the target.

So with two Kalman Filters we can give more accurate the centroid and the size of the search window for CamShift.

4 Experiments

To compare the results of the original CamShift and the proposed algorithm, we experimented on various video sequences, *Highway*, *face,* and *cup*. These video sequences have been obtained by a camera with 25 per sec frame rate. And the frame has the size 320×240. In figures 2, 3, and 4 the first

row is the result of the original CamShift and the second row is the result of the proposed algorithm.

In the *Highway* sequence, the car is difficult to distinguish from the background. It moves rapidly and is small so the displacement of this target is rather large. This large displacement results in the tracking failure with the original CamShift tracker as can be observed in the first row of Fig. 2. But the proposed algorithm successfully tracks the car as can be seen in the second row of Fig. 2.

Fig. 2 *Highway* sequence, the frames 84, 95, 114, 131 are shown.

In the *face* sequence, the face is totally occluded and one hand with a similar hue to the face disturbs the tracking. We can see in the first row, when the paper moves away from the face, the original CamShift loses the face and tracks the hand instead. However the proposed algorithm can recover from the total occlusion as can be observed in the second row of Fig. 3 because of the prediction of the Kalman Filter.

In the *cup* sequence, the shape and orientation of the cup are changing. The proposed algorithm can track better than the original CamShift when the cup moves with different orientations.

Fig. 3 *Face* sequence, the frames 30, 70, 166, 178 are shown.

Fig. 4 *Cup* sequence, the frames 68, 87, 106, 138 are shown.

5 Conclusion

Target tracking in a cluttered environment remains a challenging research topic. In this paper we propose an improved CamShift algorithm. Firstly, a background-weighted histogram is introduced, so the target can be easily distinguished from the background and other targets. Secondly, the window size is calculated to track a target accurately when the target's shape and orientation are changing.

Finally, to avoid being trapped by a local maximum, we search the true maximum beyond the local one by using the Kalman Filter. By combining the CamShift and Kalman Filter, we propose a real time tracking algorithm which copes with a temporal occlusion with a small computational cost. So the proposed algorithm enhances the robustness to occlusion, avoids being trapped by a local maximum, and it can track the target accurately despite its shape and orientation change. Compared with the original CamShift algorithm, the improved CamShift algorithm shows its superior performance in various video sequences.

To further enhance the capabilities of the tracker, future work includes investigating a new target representations scheme with spatial information. Other discriminative features will be adopted for better localization and tracking performance, rather than relying solely on simple color histograms. Also we will consider adding illumination adaptation modules into the current framework to provide an even more robust tracking algorithm.

References

[1] R.T. Collins, A.J. Lipton, T. Kanade, "A System for Video Surveillance and Monitoring", *American Nuclear Society Eight Intern. Topical Meeting on Robotics and Remote Systems,* 1999.

[2] PARK, S. AND AGGARWAL, J. K. 2004. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimed. Syst.* 10, 2, 164–179.

[3] Yaakov Bar-Shalom and Thomas E. Fortmann. 1988, Tracking and Data Association. *Academic Press*, 1988.

[4] Simon J. Julier. and Jeffrey K. 1997 "A new extension of the Kalman filter to nonlinear systems," *in Proc. SPIE* ,Vol. 3068, p. 182-193

[5] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "A color-based particle filter," *in Proc.of the 1st Workshop on Generative-Model-Based Vision*, June 2002, pp. 53 – 60.

[6] R. T. Collins. Mean-shift blob tracking through scale space. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2003.

[7] G. R. Bradski. "Computer vision face tracking for use in a perceptual user interface", *Intel Technology Journal*, 2nd Quarter, 1998.

[8] Z. Zivkovic and B. Krose. An EM-like algorithm for colorhistogram-based object tracking. *In CVPR*, 2004.

[9] Nouar, O.-D.; Ali, G.; Raphael, C.2006,Improved Object Tracking With Camshift Algorithm, *IEEE ICASSP 2006*, Volume 2, 14-19 May 2006 .

[10] Hongxia Chu et al. 2007, Object Tracking Algorithm Based on Camshift Algorithm Combinating with Difference in Frame, *IEEE Automation and Logistics*,18-21 Aug. 2007, page: 51-55

[11] B. Georgescu, I. Shimshoni, P. Meer, Mean shift based clustering in high dimensions:A texture classification example, in: :*IEEE Int' l Conf. on Comp.Vision*, Vol. 2, Nice, France, 2003, pp. 456 – 463.

[12] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-Based Object Tracking", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.25, No. 5, 2003.

[13] T.Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Trans. Comm. Technology*, vol. 15, pp. 52 – 60, 1967.

[14] A.R. Smith, "Color Gamut Transform Pairs," *SIGGRAPH 78*, pp. 12-19, 1978.